WHAT IS CLAIMED IS:

1. Apparatus for transforming computer software, said computer software comprising a branch operation for conditionally bypassing a memory write operation, the apparatus comprising:

5   a. one or more computer readable storage media; and

         b. computer executable program code stored in the one or more computer readable storage media, the computer executable program code comprising:

10   i. code for creating a storage location;

         ii. code for moving said memory write operation above said branch operation;

         iii. code for selecting a target address for said memory write operation, wherein said

15   storage location is selected if said branch operation bypasses said memory write operation; and

         iv. code for replacing an original address in said memory write operation with

20   said target address.

2. The apparatus of claim 1, wherein said code for selecting said target address selects said original address if said branch operation does not bypass said memory write operation.

3. The apparatus of claim 1, wherein said code for creating, moving, selecting and replacing transforms said computer software into low level program code for performing a conditional memory write without a branch

5 operation.

4.   The apparatus of claim 3, wherein said low level program code is functionally equivalent to said computer software.

5.   The apparatus of claim 3, wherein said low level program code comprises Assembly code.

6.   The apparatus of claim 3, wherein said low level program code comprises machine language code.

7.   The apparatus of claim 1, wherein said computer executable program code comprises a compiler.

8.   The apparatus of claim 1, wherein said computer executable program code comprises an assembler.

9.   The apparatus of claim 1, wherein said code for creating said storage location comprises code for creating computer executable instructions which generate one storage location per compilation unit.

10.   The apparatus of claim 1, wherein said code for selecting said target address for said memory write operation comprises a select operation.

11.   The apparatus of claim 1, wherein said code for selecting said target address for said memory write operation comprises a conditional move operation.

12. Apparatus for transforming computer software, said computer software comprising a conditional memory write operation, the apparatus comprising:

    a. one or more computer readable storage media; and

    b. computer executable program code stored in the one or more computer readable storage media, the computer executable program code comprising:

        i. code for creating a local variable;

        ii. code for selecting either an original address of said conditional memory write operation or an address of said local variable; and

        iii. code for performing a memory write operation into said selected address.

13. The apparatus of claim 12, wherein said computer executable program code comprises a compiler.

14. The apparatus of claim 12, wherein said code for selecting bases said selection on a result of a condition in said conditional memory write operation.

15. The apparatus of claim 14, wherein said code for selecting selects said address of said local variable if said condition is not satisfied.

16. The apparatus of claim 12, wherein said computer software is written in a high level language, and wherein said code for creating, selecting, and performing transforms said computer software into low level program code for performing a speculative store.

17.   The apparatus of claim 16, wherein said low level program code is functionally equivalent to said computer software.

18.   The apparatus of claim 16, wherein said low level program code contains no control-flow dependencies.

19.   The apparatus of claim 16, wherein said low level program code comprises Assembly code.

20.   The apparatus of claim 16, wherein said low level program code comprises machine language code.

21.   Apparatus for transforming computer software, comprising:

       means for reading source code, wherein said source code includes a conditional memory write

5   operation; and

       means for generating functionally equivalent code from said source code, wherein said functionally equivalent code contains a data-flow dependency, and wherein said functionally equivalent

10   code is not fully predicated.